ACADEMIC PROGRAM

DISTRIBUTED SYSTEMS PROGRAMMING

B.F.A. IN **COMPUTER SCIENCE**

MODALITY: ON CAMPUS

ACADEMIC YEAR: 2022-2023

Name of the course:	Distributed Systems Programming
Degree :	Computer Science
Location:	Centro Universitario de Tecnología y Arte Digital
Area:	Programming
Year:	3º
Teaching period:	1
Type:	OB
ECTS credits:	6
Teaching modality:	On campus
Language:	English
Lecturer / Email	-
Web page:	http://www.u-tad.com/

SUBJECT DESCRIPTION

Area description

This subject belongs to the programming subject. This subject is dedicated to the study of programming techniques and languages on which the software engineering degree studies will be based.

Subject description

This subject aims to teach the computational principles that allow the implementation of distributed systems, in order to later implement applications that adapt as best as possible. It is planned as a laboratory where students will program applications distributed over two environments:

- Cluster-type systems, process-oriented parallel programming

- Cloud systems, service-oriented parallel programming

Students will be asked to make configurations on a computer network that will be accessed remotely to the extent possible. Applications will be implemented on this computer network that will respond to requests from remote users, which will cover different services.

To the extent possible, the services provided by Amazon AWS will be used. Specifically, at least the EC2, Lambda, RDS, E3 services can be used. Additionally, students are encouraged to explore other services available in that system, in case they can be used for their applications.

COMPETENCIES AND LEARNING OUTCOMES

Competencies

BASIC AND GENERAL SKILLSs

CG1 - Ability to understand, schedule and solve problems trough software development

CG3 - Knowledge of the scientific fundamentals applicable to the resolution of computer problems

CG4 - Ability to simplify and optimize computer systems by understanding their complexity

CG9 - Ability to learn, modify and develop new software solutions

CG10 - Use of creative techniques to carry out computer projects

CB1 That students have demonstrated knowledge and understanding in an area of study that starts from the basis of general secondary education, and is usually at a level that, although it is supported by advanced textbooks, also includes some aspects that involve knowledge from the forefront of their field of study.

CB2 Students are able to apply their knowledge to their work or vocation in a professional manner and possess the competences usually demonstrated through the development and defense of arguments and problem solving within their field of study.

CB3 That students have the ability to gather and interpret relevant data (usually within their area of study) in order to make judgements that include reflection on relevant social, scientific or ethical issues.

CB4 Students are able to convey information, ideas, problems and solutions to both specialist and non-specialist audiences.

CB5 That students have developed those learning skills necessary to undertake further studies with a high degree of autonomy.

SPECIFIC SKILLS

CE1 - Knowledge of the structure of computers, the concepts of coding, manipulation, information processing and use of low-level languages

CE7 - Knowledge of the main types of data structures and use of libraries and algorithmic techniques associated with these structures together with the complexity orders that characterize these techniques

CE8 - Conocimiento de los distintos paradigmas detrás de los lenguajes de programación/ Knowledge of the different software

paradigms that underpin programming languages

CE9 - Knowledge of effective control structures, variables, programming syntax and memory management in the development of a computer application

CE10 - Ability to work with a release manager and generate application documentation automatically.

CE15 - Knowledge of fault tolerance, adaptability, load balancing and system predictability for distributed application development

CE17 - Knowledge of the parallelization characteristics of graphics cards and high-performance architectures for application development.

CE20 - Ability to test the operation and functionality of a computer application, develop test plans and use test-oriented design and programming techniques

CE23 - Knowledge of the principles of artificial intelligence and use of deterministic search algorithms and state machines

Learning outcomes

Upon completion of the degree, the graduate will be able to:

- To understand and handle the concept of dynamic memory

- To identify classes with the relevant data of a problem

- To instance classes and objects and manage them

- To understand and use the mechanisms of inheritance, polymorphism and operator overloading

- To identify class relationships in different use-cases.

- To master an object oriented programming language.

- To master programming patterns

- To know different problem solution strategies from an algorithmic view point: divide and conquer, dynamic programming, backtracking or genetic algorithms.

- To understand algorithmic complexity, assess it and search for optimal solutions

- To code a program able to find the optimal path between any pair of nodes of a graph

- To build neural networks to solve applied problems

CONTENTS

Distributed architectures

Distributed algorithms

Fault Tolerance

Load balancing and distribution

SUBJECT SYLLABUS

Topic 1. Introduction to distributed systems

- History and beginnings
- Advantages of its use
- General introduction to cluster, grid and cloud systems

Topic 2. Distributed memory systems and Cluster

- Client/Server Computing
- Calls and remote procedures

- Message Passing
- RPC
- Clusters
- Distributed process management
- Process migration
- Mutual exclusion and deadlock

Programming in cluster-type systems (Practical topic)

- Parallel computing
- Parallel programming methodology
- Virtualization
- Message passing paradigm
  - Tools:
  - Virtualbox, Docker, kubernetes, AWS/EC2
- Practices

Topic 3. Cloud service-oriented programming

- Introduction
- Cloud Fundamentals
- Cloud Architecture
- Example of a Cloud:
  - Amazon Web Services
  - EC2
  - S3
  - Lambda
  - RDS
- Practices:
  - Programming using AWS services
  - Service implementation

TRAINING ACTIVITIES AND TEACHING METHODOLOGIES

TRAINING ACTIVITIES

LEARNING ACTIVITIES	Total hours	Hours of presence
Theoretical / Expository classes	35,64	35,64
Practical classes	18,91	18,91
Tutorials	4,00	2,00
Independent study and autonomous work of the student	51,82	0,00
Elaboration of work (group or individual)	33,82	0,00
Evaluation Activities	5,82	5,82
TOTAL	150	62,37

Teaching methodologies

Expository method or master lesson

Case learning

Learning based on problem solving

Cooperative or collaborative learning

inquiry learning

Flipped classroom methodology

Gamification

Just in time Teaching (JITT) or classroom on time

Expository method or master lesson

Case method

Learning based on problem solving

Cooperative or collaborative learning

inquiry learning

Flipped classroom methodology

Gamification

TEMPORAL DEVELOPMENT

DIDACTIC UNITS / TOPICS TIME PERIOD

Topic 1. Introduction to distributed systems Week 1

Topic 2 (first part). Distributed memory systems and Cluster Week 2

Introduction. Programming based on message passing Week 3 and 4

Practice 1: Configuration of a virtual cluster (EC2 Network + Distributed Application) Week 5 and 6

Topic 2 (second part). Distributed memory systems and Cluster Week 7-8

Practice 2: Programming remote processes (Kubernetes+Docker) Weeks 9-10

Topic 4. Grid and Cloud computing Week 11-12

Practice 3: Service-oriented programming (AWS/Cloud) Week 13-15

EVALUATION SYSTEM

ASSESSMENT SYSTEM	MINIMUM SCORE RESPECT TO THE FINAL ASSESSMENT (%)	MAXIMUM SCORE RESPECT TO THE FINAL ASSESSMENT (%)
Assessment of participation in class, exercises or projects of the course	0	30
Assessment of assignments, projects, reports, memos	30	80
Objective test	10	60

GRADING CRITERIA

ASSESSMENT SYSTEM	ORDINARY EVALUATION	EXTRAORDINARY EVALUATION
Assessment of participation in class, exercises or projects of the course	10	0
Assessment of assignments, projects, reports, memos	60	70
Objective test	30	30

General comments on the evaluations/assessments

Evaluation in Ordinary call

• The evaluation of participation in class, in practices or in projects of the subject will be carried out based on attendance and active participation in class and in the rest of the activities developed during the course. In addition, the delivery of a research work with oral presentation will be requested at the end of the course. This aspect (average attendance and presentation of the work) will represent 10% of the final grade for the subject in the ordinary call (mandatory to qualify for enrollment). Since these deliveries must be defended in class, THEY CANNOT BE RECOVERED IN EXTRAORDINARY.

• Summarizing the previous point:

o 10% of the course grade is the average of attendance/participation and completion of the presentation work (if the work is not done, the attendance/participation will not be added)

o This research work can only be submitted in an ordinary call

o It must be defended/presented in class

o The note is saved for extraordinary

o An extraordinary delivery cannot be made (that 10% would be lost if it is not achieved in ordinary, maximum grade of 9)

• Throughout the course, MANDATORY practices will be proposed that must be delivered before the indicated date through the virtual platform. These works must be defended orally (in person or via remote connection) and will account for 60% of the final grade for the subject in the ordinary call.

• Each of the practices must be defended on the dates set by the teacher. All mandatory practices must be evaluated with a minimum of 50% of the maximum grade in order to pass. Those students who do not pass this grade or who decide to voluntarily discard it must make a complementary submission with the changes proposed by the teacher.

• To pass the subject in the ordinary call, it is essential that the final grade (including the partial exams, the practices to be delivered and participation) is at least 5.0 (out of 10). In addition to this requirement, it is necessary that the average of the exams be at least 5.0 (out of 10), and that all practices have a minimum grade of 5.0. In summary, you must have a minimum grade of 5.0 in all practices and exams to pass. If any of these requirements are not met, the subject will be automatically considered failed regardless of the rest of the grades.

• Research work can only be submitted in the first call. This means that, once the submission date has passed, there will be no possibility of delivering it later. It is not mandatory to carry out research work and oral presentation to pass the subject. If they do not do it, the student will give up 10% of the grade, with their maximum grade being a 9 in the subject, assuming that they had a 10 in the rest of the evaluable activities and exam. In this case, it is still mandatory to obtain a minimum of 5 out of 10 in each of the practices delivered and in the exam.

Evaluation in Extraordinary call

• If the student does not pass the ordinary session in January, he or she may take the extraordinary session in July. The notes of the practical deliveries and exams carried out up to that moment will be saved. The calculation of the grade will be carried out in the same way as in the first call.

• In extraordinary calls you can recover practices and exams, but you cannot recover the research work. If the research work has not been delivered in the first call, the grade will be calculated between the work delivered and the exam, taking into account the conditions of the previous points (minimum grade of 5 in each of the practices and exams).

• The use of notes or programmable scientific calculators is not allowed in the exams, for which the student must refer to the teacher's specific instructions on this topic.

• All code and work submitted by students must be ORIGINAL. This means that they must have been developed by the students throughout the course, without external help. If code/libraries external to what is provided by the teacher are used, it must be duly documented and justified. It is allowed to consult documentation external to the subject, but the code delivered by the student must respect current copyright laws and software licenses. In any case, the student must be able to explain the code used and delivered during the course.

• Late delivery of assignments: Assignments delivered after the deadline will have a penalty in the maximum grade. This penalty will be communicated in their statements.

Copies between works: A working copy will be understood as those projects that contain equal or very similar parts, that do not comply with the rules established in the previous paragraphs. Copies of work will lead to complete suspension of the subject, with no possibility of recovery in the current course. It will be the teacher who decides the seriousness of the copy, and the final decision may be consulted and revoked by the rest of the teaching team if a second opinion is needed.

• Except in exceptional circumstances, grades of any kind will not be kept between different academic years, nor between different calls.

General considerations about the development of classes:

• The use of mobile phones in the classroom is not allowed during the continuous evaluation period, unless expressly indicated otherwise by the teacher. Laptops may only be used for activities related to the subject. The teacher may withdraw the right to use the computer from those students who use it for activities that are not related to the subject (checking emails, news or social networks, consulting or preparing activities for other subjects, etc.).

• It is not allowed to consume drinks or food in the classroom. The presence of any type of drink on the tables is also not permitted, even in closed containers.

• Active participation will be required from the student, necessary for the development of the classes.

• The student will be required to behave well at all times during classes. Bad behavior that prevents the normal development of the class may lead to expulsion from the classroom for a period of time to be determined by the teacher.

LIST OF REFERENCES (BOOKS, PUBLICATIONS, WEBSITES):

Basic Bibliography:

• Burns, Brendan. Designing Distributed Systems: Patterns and Paradigms for Scalable, Reliable Services. "O'Reilly Media, Inc.", 2018.

• Wittig, Michael, Andreas Wittig, and Ben Whaley. Amazon web services inaction. Manning, 2016.

• Golden, Bernard. Amazon web services for dummies. John Wiley & Sons, 2013.

Recommended Bibliography:

• MySQL Manuals https://dev.mysql.com/doc/

REQUIRED MATERIALS, SOFTWARE AND TOOLS

Type of classroom
Theory classroom

Board and projection system

Materials:
Personal computer with Windows, Linux or OSX (at least 8Gb ram and 50GB free HDD)

Software:
Recomendado, instalación física de Linux (No virtualizada):

- Ubuntu 20.04 como mínimo (.iso para instalar desde VirtualBox)

- QTCreator u otro IDE de programación

- Librerías gcc, g++

- Acceso a cuenta AWS (preferible AWS Educate)